EL427262615

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

09458396.120999

Client Side Caching of Printer Configuration

Inventor(s):

Tad Brockway
Madan Appiah

ATTORNEY'S DOCKET NO. MS1-436US

TECHNICAL FIELD

This invention relates to server-client systems and, in particular, to a server-client system in which system configuration information for a server is stored on a client.

BACKGROUND

In a typical server-client system, a server computer is connected to several client computers or terminals via a network. In such server-client systems, applications are executed at the server rather than at the client. The server transmits display information to the client for display at the client, and user inputs (e.g., via keyboard or mouse) at the client are transmitted to the server for processing by an application(s) executing at the server. The "client" may be an entire client computer, or alternatively an application executing at the client computer.

A server in such a system typically has an operating system that can run several client sessions concurrently. Each client user has access to various resources of the server, including the processor, data storage, application programs, etc. Software applications that are resident on the server are available to each client for independent execution by the client. Each session is independent from the other sessions and, therefore, one client cannot access information relating to another client. In this manner, the server provides a logically independent machine for each client connected to the server.

Client users frequently use peripheral devices at the client machines. For example, a user may wish to attach a printer to the user's client computer (a "local" printer) in order to print data generated by an application that is running on the

server computer. To do this, the user must manually install the local printer and redirect the printer queue created by the server to the I/O port of the client computer to which the printer is connected. Such manual installation of peripheral devices is undesirable because it requires significant time and effort on the part of the user.

A co-pending application (U.S. Patent Application No. _____) entitled "Automatic Detection And Installation Of Client Peripheral Devices By A Server", to Tad Brockway, Madan Appiah, Adam Overton, and Ritu Bahl, filed concurrently herewith, describes a system in which peripheral devices attached to the client are automatically detected and corresponding device drivers installed at the server.

However, not all client peripheral devices can be automatically detected and installed. For example, if a printer is an older model that was manufactured before automatic detection technology was developed, it will not generate an identifier signal that is recognizable by the client or the server. Or, if the attached printer is a newer model but the client runs an older operating system that does not recognize the newer model printer or that does not have automatic peripheral detection capability, such as Windows® 3.1, it is necessary for the user to manually install the printer on the server.

A major drawback of this situation is that the printer must be installed each time the client connects to the server. For example, if a user installs a client printer on a server and subsequently disconnects the client from the server, the server configuration information relating to the printer is deleted. When the user reconnects to the server, the user must manually install the same printer again.

Adding to this inconvenience is the fact that if the user desires to operate the printer utilizing printer parameter settings that are different from the default settings, the user must also change the parameter settings each time the user connects to the server. For instance, if the default paper tray is the letter-size tray but the user always uses the legal-size paper tray, the user will have to set the paper tray parameter every time the user connects to a server and installs the printer.

The technology described below addresses these disadvantages, as it provides for client-side caching of server configuration information as well as peripheral device parameter settings. When the client is reconnected to a server after having been disconnected from a server session, the cached information is automatically transmitted to the server, where it is restored. The client user is thus relieved of the burden of having to manually install the printer or reset its parameter settings every time the client is connected to a server.

SUMMARY

A server-client environment provides for client-side caching of information related to a peripheral device so that the client may automatically restore the information to a server upon connection or reconnection with the server. Peripheral devices that may be connected to the client include printers, scanners, card readers, zip drives, etc. For discussion purposes, reference herein will be made to a printer. After the printer is initially configured on the server, subsequent installations are automatic. The client user is thus relieved from having to manually install the printer each time the user connects to a server.

It is noted that the term "install" has different meanings in the art. In a hardware context, "install" refers to physically connecting a peripheral device to a computer. In a software context, "install" refers to installing the software related to a connected peripheral device. If the peripheral device is a printer, a printer driver is installed and a printer queue is created. Unless otherwise noted, the term "install" as used herein means installing software that is related to a connected peripheral device. The peripheral device is physically connected to the client, but the software related to the peripheral device is installed on the server.

In a server/client environment, there may be no means for automatic detection and installation of peripheral devices, such as a printer, that are attached at the client. If so, then a client user must manually install a printer driver to render the printer operational. The user interaction required to install and configure the printer on the server are performed at the client, but the printer is installed on the server.

After the printer is manually installed on the server from the client, the portion of the server configuration that relates to the printer is sent to the client, where it is stored. In addition, printer parameter settings are sent to the server where they are returned to the client for storing with the server configuration information. When the server/client session in which the printer has been installed is terminated, the printer configuration on the server is deleted.

It is also noted that the technology as described herein may be implemented in a server/client system in which client peripheral devices are automatically detected and installed according to the method described in the above-referenced co-pending application. Utilizing the present technology in such a system provides the advantages described herein.

Subsequent installations are automatic, regardless of whether the client is connected to the same server or to a different server, or whether the initial installation was manual or automatic. When the connection is established, the client sends the cached server configuration and printer settings data to the server. The server utilizes this data to restore the printer configuration and printer settings to the state in which they were in prior to the termination of the session. The process is performed automatically, requiring little, if any, user interaction. Therefore, after the initial installation, the client user is provided with automatic printer installation on the client for that particular printer, and the previous printer settings are automatically restored. In addition, the installation and restoration are automatic even if the client (and printer) connects to a different server.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings. The same numbers are used throughout the figures to reference like components and/or features.

Fig. 1 is a diagrammatic illustration of a server/client system.

Fig. 2 is a block diagram of a server/client system.

Fig. 3 is a flow diagram of a method for client-side caching of server configuration information and automatic installation of a peripheral device.

Fig. 4 is a flow diagram of a method for client-side caching of printer parameter settings and updating same.

DETAILED DESCRIPTION

Fig. 1 shows a server/client computer system 30 having a server 32, a first client 34 and a second client 36 interconnected via a network 38. The server 32 and the clients 34, 36 have modems or network cards, which facilitate access (direct or otherwise) to the network 38 so that the server 32 and the clients 34, 36 can exchange information over the network 38. The clients 34, 26 each have a desktop 40, 42 interface displayed. In this embodiment, the server 32 and the clients 34, 36 are conventional computers.

It is noted that, although the server/client computer system 30 depicted in Fig. 1 has a first client 34 and a second client 36, there may be one to any number of clients connected to the server 32. The number of clients that may be connected to a server is limited only by the architecture of a particular server.

The server 32 is configured to provide a logically independent machine for each client 34, 36 connected to the network 38. That is, the server 32 establishes a session for each client 34, 36, provides the desktop 40, 42 for each client 34, 36, and makes server resources available to each client 34, 36. Such resources include, but are not limited to, allocations of processor time, memory, data storage, video processing, application programs, etc. A user of either of the clients 34, 36 interacts with the desktop 40, 42 on the client 34, 36 to run software applications that reside on the server 32. While the user provides input to and receives output from the client 34, 36, most processing is performed at the server 32.

The network 38 provides a communications link between the server 32 and the clients 34, 36 through which data is transmitted. The network 38 may be a local area network (LAN), a wide area network (WAN), the Internet, or the like, provided that it can accommodate server/client functionality.

Fig. 2 shows a server/client system 50 having a server 52 and a client 54. The server 52 includes a processor 56 and memory 58. The server 52 also has a network port 60, which facilitates access to the network 38. The network port 60 may be implemented as a modem, network card, or other device that interfaces the server 52 to the network 38.

The server 52 can be implemented as a common personal computer or other type of computer, such as a laptop computer, etc. The server 52 runs an operating system 62 which is stored in memory 58 and executes on the processor 56. The operating system 62 is a multitasking operating system such as a Windows® brand operating system from Microsoft Corporation (e.g., Windows® 98, Windows® 95, Windows® NT, or other derivative of Windows®). However, other operating systems may be used.

The server 52 has a printing subsystem 64 implemented in the operating system 62 stored in memory 58. The printing subsystem 62 is used to direct all operations involving printers and printing, including installing printers, creating and managing printer queues, removing printers, uninstalling printers, etc. The printing subsystem includes a configuration tracking unit 66 that is configured to monitor server configuration with regard to client peripheral devices that are installed on the server 52. This aspect of the printing subsystem 64 will be discussed in greater detail below.

The server 52 is shown having a printer driver 70 and a printer queue 72 installed and resident within the memory 58. It is noted, however, that the server 52 will only include the printer driver 70 and the printer queue 72 when a printer has been connected to the system 50 and installed on the server 52. The printer driver 70 is a printer-specific software program that provides an interface between a printer and the server 52 and allows the server 52 to provide print functions via a printer. When the printer driver 70 is installed on the server 52, the printer queue 72 is created. The printer queue 72 accepts print jobs from the server 52 and queues them for printing. It is noted that if the peripheral device is a device other than a printer, the printer driver will be a device driver for that peripheral device, if a device driver is required by the peripheral device.

The server 52 may also include a redirected printer port 74. The redirected printer port 74 is a virtual port created by the server 52 in the event that the server is unable to identify and install a compatible driver for a client-side printer. The redirected printer port 74 is exposed by the server and can be attached to a printer queue using standard printer queue creation techniques.

It is also noted that, although one printer driver 70 and one printer queue 72 are shown in the memory 58, there may be more than one printer driver or more than one printer queue 72 stored in the memory 58—corresponding to different client sessions. Also, the printer driver 70 and the printer queue 72 shown are accessible only for the session established for the client 54. The memory 58 may contain other printer drivers and printer queues that are installed for other sessions executing on the server 52. Printer drivers and printer queues installed for other sessions on the server 52 are not available for access by the client 54.

The client 54 includes memory 76, a processor 78, and an I/O port 80 having a printer 82 connected thereto. Although the client 54 is shown as having only one I/O port 80, it is noted that the client 54 may include more than one I/O port. The printer 82 has multiple printer settings 84, such as paper size, paper orientation, printing quality, etc. Printer settings 84' is a copy of the printer settings 84 that are stored in the memory 76 of the client 54 via a method that will be discussed in detail below. The memory 76 also stores server configuration information 86. The server configuration information 86 is data monitored by the configuration tracking unit 66 of server 52.

As previously noted, the configuration tracking unit 66 of server 52 is configured to monitor server configuration information relating to client peripheral devices that are installed on the server 52. In the illustrated example of a printer, such configuration information includes, but is not limited to, data associated with the printer driver 70, the printer queue 72, the redirected printer port 74, the printer settings 84, the I/O port 80 to which the printer 82 is connected, etc.

The server sends this configuration information 86 to the client 54, where it is stored in memory 76. The configuration information 86 related to the

installation of the printer 82 is therefore readily available in the event that the configuration information 86 is required to be restored on the server 52. If the configuration information 86 subsequently needs to be restored on the server 52, the configuration information 86 is simply transmitted from the client 54 to the server 52.

The configuration tracking unit 66 is configured to detect new device installations, and can also detect when a peripheral device is removed from a system, and when a system user renames an I/O queue associated with a device. For example, if the printer 82 is removed from the system, or if the client user wishes to manually delete the printer queue 72 from the system 50, the client 54 sends notification to the server 52 to remove the printer 82. In response, the server removes all data structures related to the printer 82 (e.g., printer driver 70 and printer queue 72) from the server. The server will then send updated server configuration information 86 to the client 54, where it is stored in memory 76. When the updated information 86 is stored in the memory 76 of the client 54, it simply overwrites the old information.

Likewise, if a user renames the printer queue 72, the server will send updated server configuration information 86 to the client 54, where it is stored in memory 76.

Similar to the server 52, the client 54 also has a network port 88 to facilitate access to the network 38. The network port 88 may be implemented as a modem, network card, or other device which interfaces the client 54 to the network 38.

Fig. 3 is a flow diagram depicting a method for client-side caching of server configuration information and printer settings, and for automatically installing a

peripheral device upon subsequent connection to a server. The method is described with continuing reference to Figs. 1 and 2.

At step 200 in Fig. 3, the client 54 sends a request to the server 52 to install the printer 82 on the server 52. At this time, or immediately thereafter, the printer settings 84 are also sent to the server 52. When the server 52 receives the install request from the client 54, the server performs a manual installation of the printer 82 on the server 52 at step 202. Manual installation refers to a client user being required to identify the printer driver 70 associated with the printer 82 as well as the I/O port 80 to which the printer 82 is connected. The server prompts the user for this information through the I/O interface of client 54.

When the installation is complete, the configuration tracking unit 66 of the printing subsystem 64 sends server configuration information 86 to the client (step 204). At step 206, the client 54 stores the server configuration information 86 and the printer settings 84 in the memory 76.

When the client session has concluded, the client 54 is disconnected from the server 52 (step 208). Whenever the client 54 is disconnected from the server 52, all data structures associated with peripheral devices attached to the client 52 are deleted from the server 52.

At step 210, the client 54 is reconnected to the server 52. It is noted that, although the client 54 in this example is reconnected to the server 52, the client 54 may be connected to a different server (not shown) in another server/client system (not shown). The methodology is the same whether the client 54 is connected to the server 52 or to a different server (not shown). This is another advantage to the present invention in that, once the printer 82 is manually installed on the server 52,

subsequent installations – whether they be on the server 52 or on another server (not shown) – are automatic.

Upon reconnection with the server 52, the client 54 transmits the server configuration information 86 that was stored during the previous connection, to the server 52. The printer settings 84 are also sent to the server 52. At step 212, the server 52 utilizes the server configuration information 86 received from the client 54 to automatically install the printer 82 on the server 52. This automatic installation is performed with minimal user interaction.

Once the printer 82 has been automatically installed on the server, the configuration tracking unit 66 of the server 52 sends the server configuration information 86 and the printer settings 84' to the client 54 (step 214). Upon receipt of this data, at step 216, the client 52 stores the server configuration information 86 and the printer settings 84 in memory 76.

Fig. 4 is a flow diagram depicting a method for client-side caching of server configuration information and printer settings, and for automatically updating peripheral device settings on the server. The method is described with continuing reference to Fig. 1 and Fig. 2. It is noted that the method of Fig. 4 is performed contemporaneously with the method described in Fig. 3. However, for discussion purposes, the methods are described separately.

At step 300 in Fig. 4, the client 54 sends a request to the server 52 to install the printer 82 on the server 52. At this time, or immediately thereafter, the printer settings 84 are also sent to the server 52. When the server 52 receives the install request from the client 54, the server performs a manual installation of the printer 82 on the server 52 at step 302. Manual installation refers to a client user being

required to identify the printer driver 70 associated with the printer 82 as well as the I/O port 80 to which the printer 82 is connected.

When the installation is complete, the configuration tracking unit 66 of the printing subsystem 64 sends server configuration information 86 to the client (step 304). At step 306, the client 54 stores the server configuration information 86 and the printer settings 84' in the memory 76.

At step 308, the system 50 detects if the user makes any changes to the printer settings 84. If so, the changed printer settings 84 are sent to the server 52 at step 310. At step 314, the server 52 receives the updated printer settings 84. The updated printer settings 84 are sent to the client 54 at step 316. The client 54 then stores the updated printer settings 84 at step 316.

As described in the discussion for Fig. 3, the printer settings 84' are sent from the client 54 to the server 52 when the client 54 is reconnected to the server 52. By updating the cached printer settings 84' as they are updated by the user, the desired printer settings 84 are always restored when the client 54 is reconnected to the server 52.

Conclusion

The system and method as described herein provide a user with a way to automatically install peripheral devices on a server after they have been redirected on the server one time from a client. In addition, the user's device settings preferences are preserved from session to session, thereby saving the user from having to manually update these settings whenever a server-client connection is established. Once a manual installation has been performed, subsequent installations are automatic, whether the client is reconnected to the same server, or to a different server.

Although the invention has been described in language specific to structural features and/or methodological steps, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the specific features and steps are disclosed as preferred forms of implementing the claimed invention.